Patrick Freeman

N8890862

INB383

# Statement of Completeness

Things I didn’t do

# Path Finding

**A\* Search Algorithm**

* Uses a heuristic to estimate the cost to the goal
* Guaranteed to find the shortest path (admissible)
* Can take a long time on large graphs
* Good when an absolute result is needed

**Greedy Search Algorithm**

* Uses a heuristic to find the cost of choosing a node
* Uses the cheapest option first (best-first-search)
* Doesn’t always find the shortest overall path
* Very Fast

In my assignment I have used A\* path finding for one of my NPC’s. I would have liked to made the AI update its position smoothly over each frame, but due to time constraints I have left it updating one point per frame.

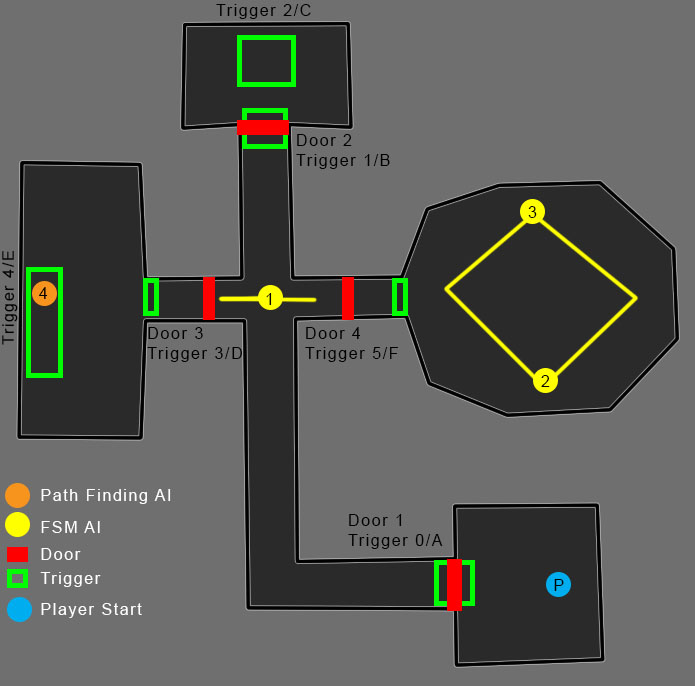
# Finite State Machines

The finite state machines I have created are used for the NPC’s and doors in my game. The door state machines are fairly simple because they only have two states, open/closed. The NPC’s state machines are a little more complex with 3 states each, with varying attributes.

Two of my doors are identical in function, so I have used the same state machine for both. Both doors will open when the player enters the trigger and close when they leave the trigger. A small issue with doing this is that they both open at the same time when their respective triggers react. This didn’t matter at all because the player will never be able to see both doors at the same time. The other two doors will stay open once a certain trigger has been activated. This unlocks the other rooms for the player.

The 3 NPC’s that are connected to state machines. Two of the NPC’s are identical and will patrol the large room by default. The hallway NPC will just patrol the cross section of the hallway. The hallway NPC has the option to patrol, wobble and emit light, and kill.

## Trigger layout



## State Diagram

# Conclusion